Űrlap létrehozása JQueryvel MVC szerkezetben

**A cél, egy olyan űrlapkezelő alkalmazás létrehozása, mely tetszőleges szerkezetű űrlap létrehozására képes.**

Nézzük lépésről lépésre!

Első lépésként olyan űrlapot fogunk létrehozni, amelyben egy **nevet** és egy **születési évet** adhatunk meg.

# Egyszerű űrlap létrehozása MVC szerkezettel

Alakítsuk ki a szokásos MVC szerkezetet!

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A Modellben majd a DataService osztály lesz, ami asszinkron módon küldi a kapott adatot az adatbázisba.

Az index.html-be csatoljuk be a jquery függvénykönyvtárt, a bootstrap cdn-t és a main.js fájlt modulárisan.



Hozzunk létre egy div-et, ahova majd az űrlapunkat elhelyezzük. A div-nek adjunk class nevet

A programunk belépési pontja a main.js. Itt példányosítjuk az UrlapControllert.

A képen szöveg, Betűtípus, sor, képernyőkép látható

Automatikusan generált leírás

## UrlapController

Itt példányosítjuk az UrlapView-t. Paraméterként adjuk át azt az elemet, ahová be szeretnénk tenni majd az űrlapunkat, azaz az .urlap div-et.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

## UrlapView

Létrehozása a szokásos módon történik.

A konstruktor megkapja példányosításkor a szülőelemet.

**3. sor** - A szülőelembe elhelyezzük a form tag-et.

**4. sor** – Készítünk egy hivatkozást a FORM taghez. Győződjünk meg róla, hogy a html-ben valóban megjelenik a jó helyena form tag! Irassuk ki a konzolra a this.formElem értékét. A FORM html tag-et kell kapnunk eredményül.

**5. sor** – Hívjuk meg a htmlOsszeallit metódust, melynek feladata a form html szerkezetének létrehozása.

Minden egyes formelem külön div-ben helyezkedik el, egy div tartalmaz egy label elemet és egy input elemet. (Bootstrap 5 osztályokkal)

Fontos, hogy a label for tagje és az input elem id és name attributumának értéke egyezzen meg.

**26-30. sor** - Utoljára adjunk egy submit gombot is a formhoz. Ezzel fogjuk elküldeni a beviteli mezőkben rögzített adatokat.

**A 31. sorban** az összeállított html-t adjuk hozzá az form taghez.

A Böngészőben láthatjuk a kész űrlapunkat, alján egy submit gombbal.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

## Submit gomb működése

Egészítsük ki a kódot a kontruktorban.

Ha a gombra kattintunk, akkor össze kell gyűjteni a formra bevitt adatokat, és el kell küldeni a szerver felé.

Lépések:

1. Létrehozom a gombhoz a "kapaszkodót"

2. Hozzárendelem az eseményt

3. Ebben az eseményben összegyűjtöm a form adatait

4. és felküldöm au UrlapController-nek

A kód 2. sorába, a kontruktor előtt hozzunk létre egy privát adattagot, amiben a form adatokat fogjuk tárolni:

#formAdat={}

Egészítsük ki a akódot a konstruktorban az alábbiak szerint:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## Hogy kapjuk meg az UrlapControllerben az űrlap adatait?

Ehhez létre kell hoznunk egy saját eseményt az UrlapView-ban. A saját eseményünk neve ujAdatHozzaAdasa lesz. Akkor fog kiváltódni, amikor rákattintunk a submit gombra és összegyűjtöttük az adatokat az űrlapról.

1. Hozzuk létre a trigger metódust a szokásos módon, mely létrehozza az új eseményt.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Az esemény átad egy objektumot, melynek a detail kulcsa tartalmazza azokat az információkat, melyet át akarunk adni az eseményyel. Jelen esetben az a formról összegyűjtött adatok lesznek.

Híjuk meg a triggert a submit gomb kattintás eseménykezelőjében, a formadatok összegyűjtése UTÁN:

  this.trigger("ujAdatHozzaAdasa")

## Ezután az UrlapControllerben iratkozzunk fel az eseményre.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

**Tulajdonképpen, ha eddig eljutottál, már létre tudsz hozni egész használható kis űrlapokat, kézzel.**

Még ha nem is túl elegáns megoldás, működőképes lehet, ha beépíted hozzá a validációt is.

Azonban, ha elegánsabban akarsz dolgozni, akkor érdemes egy olyan űrlapkezelőt készíteni, ami tetszőleges űrlapot fel tud építeni.

# Űrlapelemek létrehozása az adatLeiras objektum segítségével.

## Adatleíró objektum

Nézd meg, hogy a htmlOsszeallit metódusban, amikor létrehoztuk a két űrlapelemet, gyakorlatilag kétszer leírtuk csaknem ugyanazt. A két txt+= rész csak néhány nagyon jól eghatározott részben különbözik, mégpedig az attributumok értékében. Az attributumok értékét pedig az adott form elem tulajdonságai határozzák meg. Ezeket a tulajdonságokat az adatbázis táblájának mezői határozzák meg.

Nem lehetne ezt valahogy ciklusba tenni?

Lehet!

Hozzunk létre ehhez egy új fájlt, ami tartalmazni fogja az űrlapmezők leírását. Hazsnáltunk már ilyet a múlt alkalommal a táblázat fejlécének meghatározásához. Így volt ez a múlt alkalommal:

A képen szöveg, Betűtípus, képernyőkép, sor látható

Automatikusan generált leírás

Most ezt a fejlécet fogjuk kiegészíteni. Így:

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

## Űrlapelemek létrehozása az adatLeiras objektum segítségével.

Most térjünk vissza az **UrlapView**-hoz.

Importáljuk be ide az adatLeiras objektumunkat, és a htmlOsszeallit metódust módosítsuk úgy, hoyg egy ciklussal menjünk végig az adaleiras objektumon.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Automatikusan generált leírás

## De a number és text típus nem teljesen egyezik meg!

Ez így van. A két típus között nem túl nagy a különbség, csupán a text elemnél egy reguláris kifejezéssel tudjuk ellenőrizni a helyes adatbevitelt, míg a number típusnál min és max értékeket kell megadni.

De ha radio, textbox, vagy lenyíló elemet fogunk készíteni, az még jobban fog különbözni ettől a két input elemtől. Ezért érdemes létrehozni külön metódusokat a különböző típusú input mezők számára, és az a htmlOsszeallit metódusban egy elágazásban a megfelelő metódust meghívni.

Most így módosul az UrlapView osztályunk.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

## Mi lesz a validációval?

Ezek a metódusok összeállítják az elemei input mezőket. e ezek a mezők „viselkedni” is tudnak, azaz lehet valid-ak, vagy nem validak, illetve meg kell tudják mondani a saját értéküket. Ekkor pedig már érdemees egy – egy saját osztály létrehozni midnen egyes input elem számára.

# TextUrlapView osztály

Feladata a text típusú űrlap elemek hozzáfűzése a formhoz, illetve ezek értékének visszaadása, valamint a validáció elvégzése.

Az új könyvtárszerkezet:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## Az UrlapView-ban pedig az elágazásban példányosítani fogjuk ezt az osztályt.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

## A form adatok összegyűjtése is módosul

A TextUrlapView-ban létrehoztunk egy getValue() metódust, mely visszaadja az input elem aktuális értékét.

Az UrlapView-ban létrehozunk egy listát, amiben az \_\_\_UrlapView példányokat fogjuk tárolni

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Ebbe a listába kell beletenni az osztálypéldányt példányosításkor.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Amikor a submit gombra kattintunk, az formAdat összegyűjtéséhez ehhez a listán kell végigiterálni. és a getValue tagfüggvényt meghívni.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

# Validáció

A text mező validálása jellemzően reguláris kifejezéssel történik. A reguláris kifejezés egy olyan karaktersorozat, amely leírja, hogy milyen mintára kell illeszkednie a szövegnek. Pl.: Nagybetűvel kezdődjön, legalább 3 karakter legyen, nem lehet benne szám, stb.

Ezt határozzuk meg az adatLeiro pattern mezőjében.

Ezt használjuk fel a validációnál.

Az input elemünk keyup (vagy még jobb az input) eseményénél leellenőrizzük, hogy az a bevitt érték valid-e?

Most így néz ki a a kódunk. Egyelőre csak a konzolra írtuk ki, hogy a beírt adat valid-e, vagy sem.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Utoljára készítsünk még egy gettert, mely visszaadja a valid változó értékét. Ezt is le tudjuk kérni majd az UrlapView-ban.

## Validáció ellenőrzése az UrlapView-ban

Az UrlapView-ban létrehozok egy új változot: a urlapValid logikai változót. Értéke kezdetben true lesz

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Az órai megoldás itt érhető el: <https://github.com/csefikatalin/eSZF2_urlap.git>